FTML Project

nelson.vicel-farah, antoine.zellmeyer

June 2022

1 Bayes estimator and Bayes risk

Question 1

- Input space $\mathcal{X} = [0; 1]$
- Output space $\mathcal{Y} = \mathbb{R}^+$
- X uniform continuous distribution on $\mathcal{X}$
- $l(x, y) =$ squared loss
- Y $\sim$ Exp(1 + X)

The Bayes estimator in respect to the squared loss is $f^*(x) = E[Y|X = x]$, so

$$f^*(x) = E[Y \sim Exp(1 + x)]$$

We know that $E[X \sim Exp(\lambda)] = \frac{1}{\lambda}$ so

$$\mathbf{f^*(x) = \frac{1}{1 + x}}$$

And the Bayes Risk is

$$R^* = E[l(Y, f^*(X))]$$
$$= E_X[E_Y[(Y - f^*(X))^2|X]]$$
$$= E_X[Var(Y|X)]$$

We know that $Var[X \sim Exp(\lambda)] = \frac{1}{\lambda^2}$

$$= E_X[\frac{1}{(1 + X)^2}]$$
$$= \int_0^1 \frac{1}{(1 + x)^2} dx = \left[-\frac{1}{1 + x} \right]_0^1$$
$$= \mathbf{\frac{1}{2}}$$

Question 2

Let's define $\tilde{f}$ as the OLS estimator, which means

$$\tilde{f} = \begin{cases} \mathcal{X} \to \mathcal{Y} \\ x \to \hat{\theta}_1 x + \hat{\theta}_2 \end{cases}$$

With $\hat{\theta}_1$ and $\hat{\theta}_2$ the scalar parameters that minimize the squared loss. We deduce that

$$\begin{aligned}
\hat{\theta}_1 &= \frac{Cov(X,Y)}{Var(X)} \\
&= 12 * Cov(X,Y) \\
&= 12\left(E[XY] - E[X]E[Y]\right) \\
&= 12\left(E[E[XY|X]] - E[X]E[E[Y|X]]\right) \\
&= 12\left(E[\mathbf{XE[Y|X]}] - E[X]E[E[Y|X]]\right) \\
&= 12\left(E[\frac{X}{1+X}] - E[X]E[\frac{1}{1+X}]\right) \\
&= 12\left(1 - log(2) - \frac{1}{2}log(2)\right) \\
&= \mathbf{12 - 13log(2)}
\end{aligned}$$

And

$$\begin{aligned}
\hat{\theta}_2 &= E[Y] - \hat{\theta}_1 E[X] \\
&= E[E[Y|X]] - \hat{\theta}_1 \frac{1}{2} \\
&= log(2) - \frac{1}{2}(12 - 13log(2)) \\
&= \mathbf{\frac{15}{2}log(2) - 6}
\end{aligned}$$

We conclude that

$$\tilde{f}(x) = (12 - 13log(2))x + \frac{15}{2}log(2) - 6$$

The simulation in `part1_simulation.py` with the above settings and 10 000 samples of (X,Y) suggests that the Bayes estimator is better at estimating the setting than the OLS estimator. Probably because our model is not linear. Furthermore, the computed generalization error of the Bayes estimator converges to the value of the Bayes Risk we found previously ($\sim 1/2$) which lets us think that the simulation is a good estimation of the theoretical setup.

2 Bayes risk with absolute loss

Question 1

P(Y|X=x) where Y|X=x corresponds to an Exp(λ) continuous distribution.
The Bayes estimator for l_2 squared loss

$$f_2^*(x) = E[Y|X = x] = E[\lambda e^{-\lambda}] = \frac{1}{\lambda}$$

The Bayes estimator for l_1 absolute loss is the median of Y|X=x as seen in Question 2

$$f_1^*(x) = \frac{ln(2)}{\lambda}$$

We conclude that these Bayes estimators are not equal.

Question 2

We note $p = p_{Y|X=x}$

$$g(z) = \int_{\mathbb{R}} |y - z|p(y)dy$$

$$= \int_z^{+\infty} (y - z)p(y)dy + \int_{-\infty}^z (z - y)p(y)dy$$

$$= \int_z^{+\infty} yp(y)dy - z\int_z^{+\infty} p(y)dy + z\int_{-\infty}^z p(y)dy - \int_{-\infty}^z yp(y)dy$$

$$\frac{d}{dz}g(z) = -zp(z) - \left(\int_z^{+\infty} p(y)dy - zp(z)\right) + \left(\int_{-\infty}^z p(y)dy + zp(z)\right) - zp(z)$$

$$= \int_{-\infty}^z p(y)dy - \int_z^{+\infty} p(y)dy$$

Thus, $\frac{d}{dz}g(z) = 0$ if :

$$\int_{-\infty}^z p(y)dy = \int_z^{+\infty} p(y)dy$$

This occurs when both sides are equal to $\frac{1}{2}$, which means that the Bayes estimator $f^*(x)$ is the median. But to confirm this minimizes g(z), let's check the second derivative:

$$\frac{d^2}{dz^2}g(z) = 2p(z) > 0$$

The second derivative is always positive, so our solution is a minimum.

3 Expected value of empirical risk

Exercise

Step 1 :

$$E\left[R_n(\hat{\theta})\right] = E\left[\frac{1}{n}||y - X\hat{\theta}||_2^2\right]$$

$$= E\left[\frac{1}{n}||y - X((X^TX)^{-1}X^Ty)||_2^2\right]$$

$$= E\left[\frac{1}{n}||(I_n - X(X^TX)^{-1}X^T)y||_2^2\right]$$

$$= E\left[\frac{1}{n}||(I_n - X(X^TX)^{-1}X^T)(X\theta^* + \epsilon)||_2^2\right]$$

$$= E\left[\frac{1}{n}||X\theta^* + \epsilon - X\underbrace{(\mathbf{X^TX})^{-1}\mathbf{X^TX}}_{=\mathbf{I_d}}\theta^* - (X(X^TX)^{-1}X^T)\epsilon||_2^2\right]$$

$$= E\left[\frac{1}{n}||X\theta^* + \epsilon - X\theta^* - (X(X^TX)^{-1}X^T)\epsilon||_2^2\right]$$

$$= E\left[\frac{1}{n}||\epsilon - (X(X^TX)^{-1}X^T)\epsilon||_2^2\right]$$

$$= E_\epsilon\left[\frac{1}{n}||(I_n - X(X^TX)^{-1}X^T)\epsilon||_2^2\right]$$

Step 2 :

$$A \in \mathbb{R}^{n,m} B \in \mathbb{R}^{n,m}$$

$$tr(A^TB) = \sum_{i,j \in [1,n]\times[1,m]} a_{ij}b_{ij}$$

By applying the same rule to two identical matrices, we can conclude that :

$$A \in \mathbb{R}^{n,n}$$

$$tr(A^TA) = \sum_{i,j} a_{ij}a_{ij}$$

Step 3 :

$$E_\epsilon\left[\frac{1}{n}||A\epsilon||^2\right] = E_\epsilon\left[\frac{1}{n}\sum_{i=1}^{n}(A\epsilon)_i^2\right]$$

$$= E_\epsilon\left[\frac{1}{n}tr((A\epsilon)^T(A\epsilon))\right]$$

$$= E_\epsilon\left[\frac{1}{n}tr(\epsilon^T A^T A\epsilon)\right]$$

The trace is invariant under cyclic permutation.

$$= E_\epsilon\left[\frac{1}{n}tr(\epsilon\epsilon^T A^T A)\right]$$

By linearity of the trace and the expected value, we can push the expectation inside.

$$= \frac{1}{n}tr(E_\epsilon[\epsilon\epsilon^T]A^T A)$$

$$= \frac{1}{n}tr(\sigma^2 I_n A^T A)$$

$$= \frac{1}{n}tr(A^T A)\sigma^2$$

Step 4 :

$$A = I_n - X(X^T X)^{-1}X^T$$

$$A^T A = (I_n - X(X^T X)^{-1}X^T)^T(I_n - X(X^T X)^{-1}X^T)$$

$$= (\mathbf{I_n^T} - (\mathbf{X}(\mathbf{X^T X})^{-1}\mathbf{X^T}))^{\mathbf{T}})(I_n - X(X^T X)^{-1}X^T)$$

$$= (\mathbf{I_n} - (\mathbf{X^T})^{\mathbf{T}}((\mathbf{X^T X})^{-1})^{\mathbf{T}}\mathbf{X^T})(I_n - X(X^T X)^{-1}X^T)$$

$$= (\mathbf{I_n} - \mathbf{X}(\mathbf{X^T X})^{-1}\mathbf{X^T})(I_n - X(X^T X)^{-1}X^T)$$

$$= (I_n - X(X^T X)^{-1}X^T)^2$$

$$= I_n - 2X(X^T X)^{-1}X^T + (X(X^T X)^{-1}X^T)^2$$

$$= I_n - 2X(X^TX)^{-1}X^T + X(X^TX)^{-1}\mathbf{X^TX}\underbrace{(\mathbf{X^TX})^{-1}}_{=\mathbf{I_d}}X^T$$

$$= I_n - 2X(X^TX)^{-1}X^T + X(X^TX)^{-1}X^T$$

$$= I_n - X(X^TX)^{-1}X^T = \mathbf{A}$$

Thus : $\mathbf{A^TA = A}$

Step 5 : Conclude

$$E\left[R_n(\hat{\theta})\right] = E_\epsilon\left[\frac{1}{n}||(I_n - X(X^TX)^{-1}X^T)\epsilon||_2^2\right]$$

$$= \frac{\sigma^2}{n}tr(I_n - X(X^TX)^{-1}X^T)$$

$$= \frac{\sigma^2}{n}\left(n - tr(X(X^TX)^{-1}X^T)\right)$$

$$= \frac{\sigma^2}{n}\left(n - tr(X^TX(X^TX)^{-1})\right)$$

$$= \frac{\sigma^2}{n}\left(n - tr(I_d)\right)$$

$$= \frac{\sigma^2}{n}\left(n - d\right)$$

Thus :

$$E[R_X(\hat{\theta})] = E[E[R_n(\hat{\theta})]] = E[\frac{n-d}{n}\sigma^2] = \frac{n-d}{n}\sigma^2$$

Simulation

Step 6

$$E\left[\frac{||y - X\hat{\theta}||_2^2}{n-d}\right]$$

We recognize the fixed design risk with a subtle difference : $\frac{1}{n-d}$ replaced $\frac{1}{n}$, so we can use the formula found in Step 5 and apply this change.

$$= \frac{n-d}{n-d}\sigma^2 = \sigma^2$$

Step 7

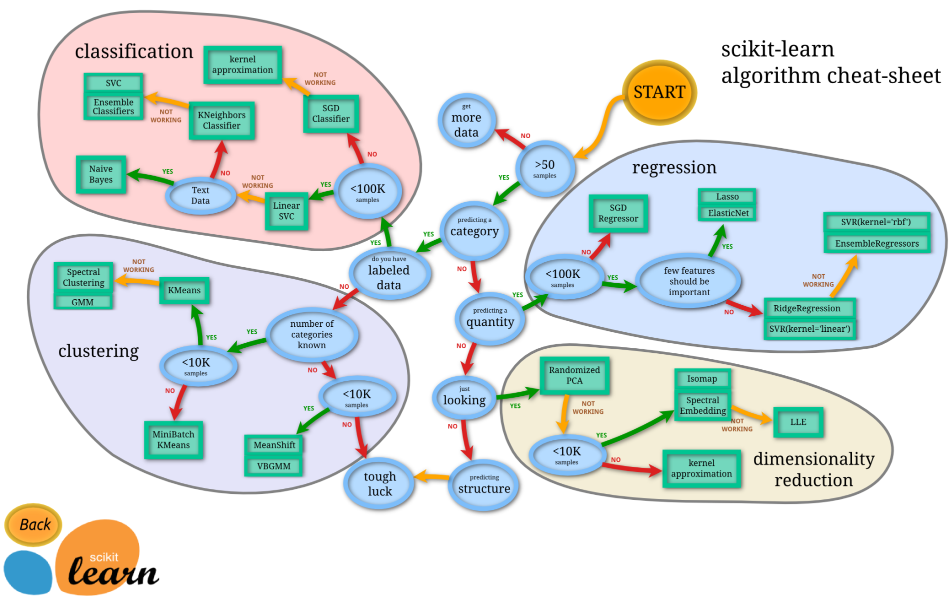
In the simulation `part3_step7.py`, we suppose :

- n = 10000
- d = 20
- $y = X\theta^*$ with θ^* a random vector

We thus estimate σ^2 with the result of Step 6. With this setting, the simulation estimates a σ^2 of 0, which is an expected result knowing that y is a simple linear function of X without any noise added to it (so the variance of ϵ is 0).
In a second setting, we define y as $y = X\theta^* + \epsilon$ with ϵ being a gaussian vector with a standard deviation of 2 and a mean of 0, for which the simulation estimates σ^2 as 4, thus being still consistent with the theoretical values.

4 Regression

For this section and the next one, we chose `scikit-learn` as our machine learning library because it contains many regression and classification models, it provides easy access to cross validation algorithms and allows us to follow this decision tree :



We should then retrieve the important properties of the dataset :
Number of entries : 1000

Number of features : 20
It means that our estimator needs to be a regression model that can handle a small set of samples and 20 features. Let's try and evaluate some of them via their cross-validation score applied on 5 subsamples of the dataset:

Model	Cross validation score
SVR Linear	0.8920
SVR rbf	0.6820
RidgeCV	**0.8932**
LassoCV	**0.8937**
ElasticNetCV	**0.8933**
RandomForestRegressor	0.8408
AdaBoost Regressor	0.8370
Gradient Boosting Regressor	0.8690

Although we should note that the cross validation score is slightly influenced by the random initialization of the models parameters, we can fairly see that **Ridge**, **Lasso** and **ElasticNet** get the best estimations of the dataset. Even though they have been already cross validated with few different hyperparameters, we can tune them even more with `GridSearchCV` from scikit-learn modules to ensure we get the most out of those estimators:

Model	Best α parameter	Cross validation score
Ridge	60	0.8933
Lasso	0.8	**0.8939**
ElasticNet	0.25	0.8935

Which lets us conclude that Lasso regression is the best model to estimate the dataset given $\alpha = 0.8$.

5 Classification

As we did in the previous section, we retrieve useful information about the dataset:
Number of entries : 1000
Number of features : 20
It means that our estimator needs to be a classification model that can handle a small set of samples and 20 features. Let's try and evaluate some of them via their cross-validation score applied on 5 subsamples of the dataset:

Model	Cross validation score
LinearSVC	**0.886**
KNeighbors	0.856
SVC	0.885
RandomForestClassifier	0.858
AdaBoostClassifier	0.864
GradientBoostingClassifier	0.858

Although we should note that the accuracy is slightly influenced by the random initialization of models parameters, the linear SVC seems to be globally the best classification model in this case.

We can finally find the best hyperparameters for `LinearSVC` the same way we did in the previous section. This lets us conclude that the best hyperparameters are : `C=1, loss=hinge, penalty=L2` as they make it possible to get a cross validation score of 0.887 (+ 0.01).